

<i>RESPONSE C</i>	
Appl. No.: 09/992,121 Office action Dated: 11/16/2006 Response Dated: 05/14/07	

Remarks/Arguments

Office Action Summary

Status.

1. This *RESPONSE C* is in answer to the Office communication mailed 11/16/2006.
2. The Office communication is non-final.
3. NA

Disposition of Claims.

4. Claims 1 and 3 - 14 are pending in the application.
5. No Claims have been allowed.
6. Claims 1 and 3 - 7 stand rejected and Claims 8 - 14 are newly added.
7. NA
8. NA

Application Papers.

9. NA
10. The drawings as previously amended have been accepted.
11. NA

Priority under 35 U.S.C. § 119.

12. NA

<i>RESPONSE C</i>	
Appl. No.: 09/992,121 Office action Dated: 11/16/2006 Response Dated: 05/14/07	

DETAILED ACTION

1. Claims 1-7 were originally presented and Claims 8 – 14 are newly added.
2. Claims 1-7 were originally rejected.
3. *RESPONSE C* was filed pursuant to a Request for Continued Examination.
4. Claim 2 was cancelled.
5. Claims 1 and 3 – 14 are currently pending in the application.

Response to Arguments

Examiner's Response: Information Disclosure Statement

6. Background is noted
7. Illegible Reference noted.
8. Examiner's Response noted.

Examiner's Response: Sections 6 through 8 of Remarks

9. Noted.

Examiner's Response: Claim Interpretation

10. The Examiner asserted a new claim interpretation as follows:

"The mere ability, enablement, or adaptability to perform a function does not necessitate the performance of such function or its use. Therefore, if a limitation does not necessitate the performance of the function, an art that does not expressly prohibit the performance of the function anticipates its performance. This interpretation is support by MPEP 2111.04 [R-31], which recites in part:"

"Claim scope is not limited by claim language that suggests or makes optional but does not require steps to be performed, or by claim language that does not limit a claim to a particular structure. However, examples of claim language, although not exhaustive, that may raise a question as to the limiting effect of the language in a claim are: (A) adapted to" or 'adapted for" clauses; (B) "wherein" clauses; and (C) "whereby" clauses." (MPEP 2111.04)

RespC_07-05-14-fi.doc		5/14/2007-2:23:30 PM
Atty Doc No: AMDH-08156USO	Page 8 of 23	Ref #:06-51554FIH

<i>RESPONSE C</i>	
Appl. No.: 09/992,121 Office action Dated: 11/16/2006 Response Dated: 05/14/07	

10.1. The Examiner's new claim interpretation is misleading and is not supported by the quotation from MPEP 2111.04. In particular, MPEP 2111.04 has no reference whatsoever to "prior art" or any "construction of prior art" and therefore MPEP 2111.04 does not support anything about the construction of prior art. Specifically, MPEP 2111.04 does not support anything as to what would constitute "anticipation" by prior art.

10.2. The Examiner's argument that "**an art that does not expressly prohibit the performance of the function anticipates its performance**" is easily demonstrated as not being true. As one example, consider an art that is incapable of performing a particular function. Consider that the art does not "expressly prohibit the performance of that particular function". Certainly, the art's mere failure to "expressly prohibit the performance of that particular function" does not convert the art into anticipating that particular function since the art is incapable of performing that particular function.

10.3. The Examiner's proposed "Claim Interpretation" is without merit. Whether a function can be performed and when and under what conditions the function should be performed must be taught by the references relied upon by the Examiner. A "Claim Interpretation" merely conjured up without any support in the references will not convert references or combinations of references that fail to teach a function into references that do teach the function.

Examiner's Response: 35 USC § 102 Rejection

11. Applicant argued in prior RESPONSE B:

11.1. "Walters ('593) only has a single block of translated instructions corresponding to a single legacy instruction.

11.2. The Examiner's reliance on the Examiner's statement 'a legacy block can consist of a single legacy instruction' cannot apply to claim 1 which now recites 'each legacy block including a plurality of legacy instructions'." (Remarks section 10.4)

RespC_07-05-14-fi.doc		5/14/2007-2:23:30 PM
Atty Doc No: AMDH-08156USO	Page 9 of 23	Ref #:06-51554FIH

RESPONSE C	
Appl. No.: 09/992,121 Office action Dated: 11/16/2006 Response Dated: 05/14/07	

12. Examiner's Response:

12.1. "Applicant's attention is drawn to (col: 6 line: 59 to col: 7 line: 7 "the dynamic recompiler 116 would handle the execution of all the instructions in non-native code applications"; col: 3 line: 64 to col: 4 line: 3)."

12.1.1. The Examiner may have misunderstood Applicant's argument. Applicant's claims require "*linking the particular translated blocks into a particular linked group*". The Examiner seemed to suggest that a single instruction could constitute a "block". To make clear that Applicant's claims were not intended to cover such a construction, the claim was amended to recite that a block included a plurality of instructions. While *Walters* '593 may handle "all the instructions in non-native code", *Walters* '593 does not do so by linked blocks. Rather, *Walters* '593 translates to a single block and then executes the single block.

12.1.2. Nothing in *Walters* '593 operates to use "*a link in each particular translated block to point to a location of the next particular translated block of the particular linked group*" as required in Applicant's Claim 1, for example. The Examiner in the Office action did not dispute Applicant's argument in prior *RESPONSE B*, paragraphs 10.1-10.4 therein.

13. Applicant statements in sections 10.5 through 10.7 of Remarks:

13.1. Applicant stated:

"The Examiner's reliance on the code chunk map 180 of FIG 2 to teach the linking of Claim 1 is not proper. In *Walters*, the chunk map 180 map functions to indicate for each page, if the cache stores at least one code block corresponding to a non-native entry point instruction in the page."

RESPONSE C	
Appl. No.: 09/992,121 Office action Dated: 11/16/2006 Response Dated: 05/14/07	

14. Examiner Response:

14.1. “The Examiner did not cite Fig 2 item 180 in the rejection of claim 1. The Examiner cited (Fig 5 step 230, col: 4, line: 35-44; the map links the translated blocks to their corresponding legacy blocks; linked group ... hash table).”

14.1.1. In *Walters* ‘593, Applicant has carefully reviewed the cited art in its entirety, not just the portions cited by the Examiner, and the FIG 2 item 180 is the only “code chunk map” in *Walters* ‘593. If the Examiner is relying on some other portion of *Walters* ‘593 for a different “code chunk map”, the Examiner is requested to make the citation on the record. The only “code chunk map” being referred to in col: 4, line: 35-44 cited by the Examiner is the Fig 2 “code chunk map” 180 discussed by Applicant.

14.1.2. Nothing in the Examiner’s citation **Fig 5 step 230, col: 4, line: 35-44** refers to any “linked group ... hash table”. The citation of the Examiner clearly does not support the “linked group ... hash table” argument of the Examiner. The **Hash Table Lookup Proc** 142 of FIG 2 finds a single **FC addr** corresponding to a single **NC addr** pointing to a single **Compiled Code Block** in Code Cache 118. There is no linked group. Rather, there is a one for one linkage from Hash Table 142 to a single Compiled Code Block in Code Cache 118. Nothing in one Compiled Code Block links to another Compiled Code Block in *Walters* ‘593. The contents of the Hash Table 142 are not in the Compiled Code Block.

14.1.3. By way of contrast, Applicant’s claims require “a link in each particular translated block to point to a location of the next particular translated block of the particular linked group”. In order for *Walters* ‘593, to have such a relationship, one or more of the Compiled Code Blocks in the FIG 2 Code Cache 118 of *Walters* ‘593 would necessarily have a link pointing to another one of the Compiled Code Blocks in the FIG 2 Code Cache 118. Nothing in *Walters* ‘593 describes any such link in any Compiled

RespC_07-05-14-fi.doc		5/14/2007-2:23:30 PM
Atty Doc No: AMDH-08156USO	Page 11 of 23	Ref #:06-51554FIH

<i>RESPONSE C</i>	
Appl. No.: 09/992,121 Office action Dated: 11/16/2006 Response Dated: 05/14/07	

Code Block. Accordingly, no linked group of Compiled Code Blocks is described in *Walters* '593. Further nothing in *Walters* '593 suggests that the sequence of executing any particular one of the Compiled Code Blocks in Cache 118 is correlated in any way to anything in another one of the Compiled Code Blocks in Cache 118.

14.2. Sections 10.5 through 10.7 of the Remarks in Applicant's prior *RESPONSE B* were noted by the Examiner and the Examiner reminded Applicant:

"To further the prosecution in a compact manner, additional portions of the reference are cited for Applicant's convenience. These citations are non-exhaustive. The Applicant is respectfully reminded to carefully review the cited art in its entirety, not just the portions cited by the Examiner."

14.2.1. Applicant has carefully reviewed the cited art and cannot find anything that supports the Examiner's arguments. If the Examiner is aware of anything that supports the Examiner's arguments, the Examiner is respectfully requested to make the citation of record.

15. Applicant argues in prior RESPONSE B (section 10.9 of Remarks):

15.1. "The linking, if any, in *Walters* is to identify pages and *Walters* has no disclosure relating to linked executable translated instructions linked in a linked group."

16. Examiner Response:

16.1. "The linking is inherent in *Walters*' disclosure. Specifically, after executing a set of instructions an entry point is detected which either points to additional native, non-native code, or non-native code that is translated and cached (**col: 7 line: 52-63 the entry point serves as a pointer to a next particular block (pre-defined set of non-native instructions); col: 5 line: 19-30; col: 7 line: 52-63).**"

16.1.1. While the Examiner has argued that some "linking is inherent in *Walters*", that linking, if it exists, is not the linking claimed by Applicant. Applicant claims "*a link in*

RespC_07-05-14-fi.doc		5/14/2007-2:23:30 PM
Atty Doc No: AMDH-08156USO	Page 12 of 23	Ref #:06-51554FIH

RESPONSE C	
Appl. No.: 09/992,121 Office action Dated: 11/16/2006 Response Dated: 05/14/07	

each particular translated block to point to a location of the next particular translated block of the particular linked group". The citation by the Examiner to **col: 7 line: 52-63** clearly does not describe or suggest such a link. The **col: 7 line: 52-63** citation by the Examiner refers to an operation that is, either (i) an instruction (one of a predefined set that only executes by interpretation) that does not execute by executing any Compiled Code Block in the Code Cache 118 or (ii) an instruction that executes by executing a single Compiled Code Block in the Code Cache 118 *Walters '593*. The **col: 7 line: 52-63** citation has no suggestion whatsoever of linking Compiled Code Blocks.

17. Applicant's section 10.11-10.13, and 11-11.2 of prior RESPONSE B:

17.1. "As per Claim 3 as discussed in Section 10.7 above and contrary to the Examiner's argument regarding Claim 3, Walters does not operate such that 'all translated instructions in the group are purged when any are purged (emphasis in original) (Section 10.11 of Remarks)."

18. Examiner Response:

18.1. "Subsection 1 of the section *supra* amounts to mere allegation of patentability. See MPEP 716.01 (c) section II."

18.1.1. The MPEP 716.01 (c) section II citation by the Examiner does not apply to the Applicant's arguments. Specifically, MPEP 716.01 (c) section II is as follows:

"II. < ATTORNEY ARGUMENTS CANNOT TAKE THE PLACE OF EVIDENCE

The arguments of counsel cannot take the place of evidence in the record. In re Schulze, 346 F.2d 600, 602, 145 USPQ 716, 718 (CCPA 1965). Examples of attorney statements which are not evidence and which must be supported by an appropriate affidavit or declaration include statements regarding unexpected results, commercial success, solution of a long-felt need, inoperability of the prior art, invention before the date of the reference, and allegations that the author(s) of the prior art derived the disclosed subject matter from the applicant."

18.1.2. The Applicant's arguments are not any of "commercial success, solution of a long-felt need, inoperability of the prior art, invention before the date of the reference, and

RespC_07-05-14-fi.doc		5/14/2007-2:23:30 PM
Att'y Doc No: AMDH-08156USO	Page 13 of 23	Ref #:06-51554FIH

RESPONSE C	
Appl. No.: 09/992,121 Office action Dated: 11/16/2006 Response Dated: 05/14/07	

allegations that the author(s) of the prior art derived the disclosed subject matter from the applicant” as listed in MPEP 716.01 (c) section II.

18.1.3. Applicant’s arguments are obviously well founded and would easily be recognized as valid by any one of ordinary skill in the art. Applicant’s argument is that “Walters does not operate such that ‘all translated instructions in the group are purged when any are purged’. Applicant’s argument is obviously true because *Walters* ‘593 only stores information relative to purging on a page basis in Code Chunk Map 180. *Walters* ‘593 highlights the page purging capabilities of the *Walters* ‘593 system and any resolution on other than a page basis would require information that is not stored in *Walters* ‘593.

18.2. Applicant has reviewed the cited art in its entirety.

19. Upon further review, Applicant believes the Examiner should find Applicant's arguments persuasive.

Claim Rejections - 35 USC § 101

20. Claims 1-6 stand rejected under 35 U.S.C. § 101 because the claimed invention is allegedly directed to nonstatutory subject matter. The Examiner for the rejection relies upon MPEP 2106 that recites, in part:

"...USPTO personnel shall review the claim to determine it produces a useful, tangible, and concrete result. In making this determination, the focus is not on whether the steps taken to achieve a particular result are useful, tangible, and concrete, but rather on whether the final result achieved by the claimed invention is "useful, tangible, and concrete." (emphasis added)

RespC_07-05-14-fi.doc		5/14/2007-2:23:30 PM
Atty Doc No: AMDH-08156USO	Page 14 of 23	Ref #:06-51554FIH

<i>RESPONSE C</i>	
Appl. No.: 09/992,121 Office action Dated: 11/16/2006 Response Dated: 05/14/07	

20.1. The Examiner argues:

“The method claims do not produce a useful, tangible, and concrete final result. The steps of the method claims do not produce a useful, tangible, and concrete final result. They merely recite a software algorithm, per se, which, for example, does not display, store, or otherwise provide a useful tangible output. Note exemplary claim 1 which only recites software steps and does not produce a useful tangible and concrete result. See MPEP 2106 [R-5].”

20.2. The Examiner’s rejection under 35 U.S.C. § 101 appears for the first time in the Office action dated 11/16/2006 and did not appear in either of the Office actions dated 07/19/2005 and 04/13/2006.

20.3. The rejection is traversed since clearly the method claims produce a useful, tangible, and concrete final result.

20.4. The method claimed is for “dynamic emulation of legacy instructions” and the method is implemented in a computer. Anyone skilled in the art would immediately recognize that such a computer-implemented “dynamic emulation of legacy instructions” produces “a useful, tangible, and concrete final result”. The final result is “emulation of legacy instructions” and no one can reasonably question that such a result is both “tangible” and “useful”.

20.5. The Examiner appears to argue that most all “software algorithms” (that is, the ones that do not provide a physical transformation) in a computer do not produce a “tangible result” and that “software steps” that perform such “software algorithms” cannot produce “a useful tangible and concrete result.” Such notions of the Examiner are clearly at odds with the reality of the computer industry and society today in which billions of dollars are expended because execution of such computer instructions in the form of software steps produce real, recognized and demanded results. Such results are not abstract, are not non-repeatable and are not lacking

RespC_07-05-14-fi.doc		5/14/2007-2:23:30 PM
Atty Doc No: AMDH-08156USO	Page 15 of 23	Ref #:06-51554FIH

RESPONSE C	
Appl. No.: 09/992,121 Office action Dated: 11/16/2006 Response Dated: 05/14/07	

in utility. If execution of computer instructions in the form of software steps were merely abstract or non-repeatable operations lacking in utility, then the huge market for such products would not exist.

20.6. Applicant's claimed method is not abstract, is repeatable and is particularly useful because it permits computer programs written with "legacy instructions" to be executed efficiently as blocks of translated instructions. Typically, the translated instructions execute on more modern computers that may not have even existed when the computer programs with "legacy instructions" were originally written.

Claim Rejections - 35 USC § 102

21. Claims 1, 3, 5, and 7 stand rejected under 35 U.S.C. 102(b) as being anticipated by Walters (US **5,768,593**).

21.1. In making the rejection the Examiner applies *Walters* '593 as follows (Section numbers added by Applicant):

21.1.1. Walters discloses: 1. A computer-implemented method for dynamic emulation of legacy instructions comprising:

21.1.2. accessing said legacy instructions in legacy blocks, each legacy block including a plurality of legacy 'instructions (**col: 4 line: 4-8 legacy blocks ... "extended block of 'qualifying' non-native code"; col: 4 line: 20-28 instructions ... codes; col: 6 line: 18-26 ... legacy code**),

21.1.3. for each particular legacy instruction in a particular legacy block,

21.1.4. translating the particular legacy instruction into one or more particular translated instructions for emulating the particular legacy instruction (**col: 7 line: 16-23; col: 7 line: 52-63**),

RespC_07-05-14-fi.doc		5/14/2007-2:23:30 PM
Atty Doc No: AMDH-08156USO	Page 16 of 23	Ref #:06-51554FIH

RESPONSE C	
Appl. No.: 09/992,121 Office action Dated: 11/16/2006 Response Dated: 05/14/07	

- 21.1.5. organizing the particular translated instructions into one or more particular translated blocks (**col: 7 line: 52-63**),
- 21.1.6. linking the particular translated blocks into a particular linked group corresponding to said particular legacy block; said linking using a link in each particular translated block to point to a location of the next particular translated block of the particular linked group (**col: 7 line: 52-63 the entry point serves as a pointer to a next particular block (pre-defined set of non-native instructions)**),
- 21.1.7. executing the particular translated instructions in the particular translated (sic, blocks) by sequentially executing the linked group translated blocks (**col: 5 line: 19-30; col: 7 line: 52-63**).
- 21.2. Applicant traverses the rejection under 35 U.S.C. 102(b) for the following reasons.
- 21.2.1. Applicant's Claim 1 by way of example requires "*linking the particular translated blocks into a particular linked group corresponding to said particular legacy block; said linking using a link in each particular translated block to point to a location of the next particular translated block of the particular linked group*" as quoted from Applicant's Claim 1 in Section 21.1.6 above where the Examiner argues that such limitation is found in **col: 7 line: 52-63 of Walters '593**. However, **col: 7 line: 52-63 of Walters '593** does not describe or suggest such a limitation.
- 21.2.2. The Examiner not only relies on **col: 7 line: 52-63 of Walters '593** to find Applicant's limitation in Section 21.1.6, the Examiner uses that citation for each of Applicant's elements in Section 21.1.4, Section 21.1.5, Section 21.1.6 and Section 21.1.7.

RESPONSE C	
Appl. No.: 09/992,121 Office action Dated: 11/16/2006 Response Dated: 05/14/07	

21.2.3. The Examiner's entire rejection hangs on the meaning of the citation **col: 7 line: 52-63** of *Walters '593* which is set forth as follows:

"If the entry point instruction in the non-native code application does not correspond to a code block in the code cache (162-N), the recompiler 116 begins recompilation of the corresponding code block. However, if during decoding of the entry point instruction by the recompiler it is determined that the entry point instruction is one of a **predefined set** of non-native instructions to be executed by an interpreter (166-Y), then that instruction is executed by the interpreter (168). Otherwise (166-N), the cross-compiler continues with compilation of a block of non-native code (170), and then the resulting native code block is executed (step 172)." (Bold added by Applicant)

21.2.4. The **col: 7 line: 52-63** citation by the Examiner refers to an operation flow depicted in FIG 3 of *Walters '593* commencing with a hash table NO output (162-N). If the FC instruction is one that only executes by "interpretation", the output (166-Y) is to the Call FC Interpreter 168 whereby further instruction execution is by "interpretation" and no Code Block at all is executed. Alternatively, if the FC instruction is one that is not one of the set designated for "interpretation", the output (166-N) is to the cross-compile code block 170. The operation is described in the Examiner's citation **col: 7 line: 52-63**, particularly, the last four lines thereof as "... *the cross-compiler continues with compilation of a block of non-native code (170), and then the resulting native code block is executed (step 172).*" Notice in that quotation that *native code block* is singular. No reference is made to plural *Code Blocks* and no reference is made to linking of *Code Blocks* anywhere in the Examiner's citation **col: 7 line: 52-63**. Furthermore, Applicant cannot find anything else in *Walters '593* that in any way teaches the linking of Code Blocks in the *Walters '593* Code Cache 118.

21.2.5. The "predefined set" as appears in the **col: 7 line: 52-63 quote** of *Walters '593* and as quoted in Section 21.2.3 above is not the same as a "block" as used in

RespC_07-05-14-fi.doc		5/14/2007-2:23:30 PM
Atty Doc No: AMDH-08156USO	Page 18 of 23	Ref #:06-51554FIH

RESPONSE C	
Appl. No.: 09/992,121 Office action Dated: 11/16/2006 Response Dated: 05/14/07	

Applicant's claims and specification. As used in *Walters* '593, "predefined set" is a category or classification of instruction types. When anyone of the the instructions of that type appears, the execution is immediately directed to "interpretation" as discussed in Subsection 21.2.5 above. In the Applicant's claims and specification, a "block" is a contiguous instantiation of instructions to be executed (i.e. in a program). To the extent that the Examiner is equating the "predefined set" of *Walters* '593 with the "block" in Applicant's claims, the Examiner is clearly in error.

21.3. All of the Claims dependent from Claim 1 are distinguished over *Walters* '593 in the same manner as Claim 1 and as discussed above in connection with the Examiner's Responses to Applicants Remarks in prior *RESPONSE B*.

21.4. The Remarks above pertaining to Claim 1 apply equally to Claim 7.

22. Claims 1 and 7 stand rejected under 35 U.S.C. 102(b) as being anticipated by Mann (US 5,768,593).

22.1. In making the rejection the Examiner applies *Mann* '593 as follows (Section numbers added by Applicant):

22.1.1. Mann discloses: 1. A computer-implemented method for dynamic emulation of legacy instructions comprising:

22.1.2. accessing said legacy instructions in legacy blocks, each legacy block including a plurality of legacy instructions (**col: 5 line: 64-63**),

22.1.3. for each particular legacy instruction in a particular legacy block,

22.1.4. translating the particular legacy instruction into one or more particular translated instructions for emulating the particular legacy instruction (**Fig 3, 4, 5, and their descriptions; col: 2 line: 44-60**),

RespC_07-05-14-fi.doc		5/14/2007-2:23:30 PM
Atty Doc No: AMDH-08156USO	Page 19 of 23	Ref #:06-51554FIH

RESPONSE C	
Appl. No.: 09/992,121 Office action Dated: 11/16/2006 Response Dated: 05/14/07	

- 22.1.5. organizing the particular translated instructions into one or more particular translated blocks (**col: 2 line: 44-60**),
- 22.1.6. linking the particular translated blocks into a particular linked group corresponding to said particular legacy block; said linking using a link in each particular translated block to point to a location of the next particular translated block of the particular linked group (**col: 2 line: 44-60; col: 6 line: 11-28; col: 6 line: 47-61**),
- 22.1.7. executing the particular translated instructions in the particular translated by sequentially executing the linked group translated blocks (**col: 6 line: 11-28**).
- 22.2. Applicant traverses the rejection under 35 U.S.C. 102(b) for the following reasons.
- 22.2.1. Applicant's Claim 1 by way of example requires "*linking the particular translated blocks into a particular linked group corresponding to said particular legacy block; said linking using a link in each particular translated block to point to a location of the next particular translated block of the particular linked group*" as quoted from Applicant's Claim 1 in Section 22.1.6 above where the Examiner argues that such limitation is found in (**col: 2 line: 44-60; col: 6 line: 11-28; col: 6 line: 47-61**) of *Mann '593*. However, (**col: 2 line: 44-60; col: 6 line: 11-28; col: 6 line: 47-61**) of *Mann '593* do not describe or suggest such a limitation.
- 22.2.2. Referring to **col: 2 line: 44-60** of *Mann '593*, it states in lines 53 – 56 that:
- "Upon completion of the Host code block, execution control is returned to the emulator, with an indication of the next Target system instruction to execute.
- 22.2.2.1. It is clear from the quote from **col: 2 line: 44-60** of *Mann '593* that control after one block is always returned to the emulator; it is not linked to another block as in Applicant's claims.
- 22.2.3. Referring to **col: 6 line: 11-28** of *Mann '593*, it states in the last five lines that:

RespC_07-05-14-fi.doc		5/14/2007-2:23:30 PM
Atty Doc No: AMDH-08156USO	Page 20 of 23	Ref #:06-51554FIH

RESPONSE C	
Appl. No.: 09/992,121 Office action Dated: 11/16/2006 Response Dated: 05/14/07	

“Control is then transferred to the Host code 88 at that point. The Host code 88 is then executed until the end of a block is encountered, at which time control is transferred back to the emulator with an indication of where to pick up interpreting code.”

22.2.3.1. It is clear from the quote of **col: 6 line: 11-28** of *Mann '593* that control after one block is always returned to the emulator; it is not linked to another block as in Applicant's claims.

22.2.3.2. Referring to **col: 6 line: 47-61** of *Mann '593*, nothing therein discusses processing at the end of a block. It therefore has no suggestion about exiting one block and directly entering a new block according to a link in the block as in Applicant's claims.

22.3. Claim 7 is distinguished from *Mann '593*, in the same manner as Claim 1.

22.4. All of the Claims dependent from Claim 1 are distinguished from *Mann '593* in the same manner as Claim 1.

23. The newly added Claim 8 is distinguished over the *Walters '593* and *Mann '593* references for all of the reasons discussed above in connection with Claim 1.

23.1. Additionally, Claim 8 recites the linked list relationship in the cache based upon Applicant's specification description at paragraph [32], for example, as follows:

said logical group including a first translated block, one or more next translated blocks and a last block, said linking using a linked list in said cache including a first link in the first translated block that points to a location in the cache of a next translated block, one or more next links in the next translated blocks where each next link points to a location in the cache of a subsequent one of the next translated blocks, and a last link that points to the last block of the logical group,

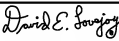
RESPONSE C	
Appl. No.: 09/992,121 Office action Dated: 11/16/2006 Response Dated: 05/14/07	

24. The newly added Claim 9, dependent from Claim 8, is distinguished over the *Walters '593* and *Mann '593* references for all of the reasons discussed above in connection with Claim 8.
- 24.1. Additionally, Claim 9 recites the *taken branch* relationship when the *branch target instruction* is within the logical group in the cache and Claim 9 is based upon Applicant's specification description at paragraph [35], for example.
25. The newly added Claim 10, dependent from Claim 8, is distinguished over the *Walters '593* and *Mann '593* references for all of the reasons discussed above in connection with Claim 8.
- 25.1. Additionally, Claim 10 recites the operation where "each logical group of translated blocks corresponds to at least one legacy block" based upon Applicant's specification description at paragraph [31], for example.
26. The newly added Claim 11, dependent from Claim 8, is distinguished over the *Walters '593* and *Mann '593* references for all of the reasons discussed above in connection with Claim 8.
- 26.1. Additionally, Claim 11 recites the operation where "each legacy block has a number of translated blocks where the number of translated blocks differs for different legacy blocks" based upon Applicant's specification description at paragraph [31], for example.
27. The newly added Claims 12, 13 and 14, dependent from Claim 8, are patterned on Claims 4, 5 and 6, dependent from Claim 1, and are distinguished over the *Walters '593* and *Mann '593* references for all of the reasons discussed above in connection with Claim 1, Claim 8 and Claims 4, 5 and 6.

RESPONSE C	
Appl. No.: 09/992,121 Office action Dated: 11/16/2006 Response Dated: 05/14/07	

28. For all of the reasons above, reconsideration of all claims now in the application is requested.

Respectfully submitted,

	SIGNATURE	
David E. Lovejoy (US Reg. No.: 22,748)	 /david lovejoy/	Signature Date 14-May-07
Mail Address	Customer No.	Communication
102 Reed Ranch Rd. Tiburon, CA 94920-2025 USA	21603	Tel: (415) 435-8203 Fax: (415) 435-8857 e-mail:david.lovejoy@sbcglobal.net

RespC_07-05-14-fi.doc		5/14/2007-2:23:30 PM
Atty Doc No: AMDH-08156USO	Page 23 of 23	Ref #:06-51554FIH